MISQ: A UML-based Analytical Modeling Methodology for Optimizing Web Service Composition

SeogChan Oh
Dongwon Lee
Soundar Kumara

Penn State University, USA

March 29, 2005

Outline

- What is Web Services?
- Motivation
- Main Idea: MISQ
- Illustration
- Conclusion
SOA: Service-Oriented Network

- Services? “A procedure, method, or object with a stable, published interface that can be invoked by clients”
- Popular means to establish services is Web Services

Web Services

- “Web Services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes...Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service…” [IBM]
Web Services

- XML-based framework for machine interoperability
  - API: WSDL
  - Communication: SOAP
  - Yellow Page: UDDI
- Disguised RPC or CORBA in XML
- Let software agents communicate each other without human intervention (in theory)

Big Picture

- C++ Client
- Java Client
- Perl Client
- Python Client

Publish

Find

Invoke
Web Services

- Web services market will be $21 billion by 2007 and will peak at $27 billion in 2010 [IDC, 2003]
- 80% of US enterprises will have some type of Web services project under way by 2008 [IDC, 2003]
- 41% percent of enterprise software purchased in 2007 will be Web-services-enabled [Gartner, 2004]
- Tools for Web Services are needed to:
  - Discover
  - Compose
  - Analyze
  - Optimize
  - …

Web Services Research @ Penn State, USA

- **Atherton** project is to develop tools/methodology for Web Services
  - [http://nike.psu.edu/atherton/](http://nike.psu.edu/atherton/)
- Sub-projects
  - Analysis & Optimization: **MISQ** [BSN 05]
  - Discovery and Composition: **BF*** [IEEE EEE 05, Microsoft SciData Grant 05]
  - Generation: **Pollock** [ACM SAC 05, ECDL 04, IBM Eclipse Grant 04]
Motivation

- How many loan companies are appropriate?
- How much charge to customers is appropriate?
- ...

Motivation

- Composing various services in various ways can yield value-added services in BSN
- Finding an optimal configuration in general settings is *NP-complete* (proof via a reduction to SAT problem)
- Nevertheless, optimization is still feasible for a small-size setting => How?
  - People start from graphical model: UML, ER, …
  - Other mathematical models are good for analysis: Petri Net, Automata, …
Overview of MISQ

1. Design high-level UML diagrams such as state and sequence diagrams
2. Transform high-level UML designs into a formal model in Stochastic Process Algebra (SPA) and Generalized Stochastic Petri-Net (GSPN) models
3. Analyze and optimize it
4. Based on the optimized high-level design, generate high-fidelity UML
5. From the high-fidelity UML, generate implementation artifacts.
SPA & GSPN

- SPA (Stochastic Process Algebra)
  - Popular formal framework to model processes
- GSPN (Generalized Stochastic Petri Net)
  - Popular graphical stochastic framework to model processes
  - Easy manipulation in simulations
  - Many tools supporting web services already
- Conversion of flow
  - UML => SPA => GSPN

### SPA => GSPN

**SPA**

```
choice_decision := (accept + reject).
```

**GSPN**

![GSPN Diagram](image-url)

- `choice_start`
- `accept`
- `reject`
- `accept_decision`
- `reject_decision`
Illustrative Example

1. C seeks for an auto loan with minimum interest rate, sends an inquiry to B (C has no direct access to WS).
2. B relays the C’s request to each ws, ∈ WS.
3. ws calculates and returns its Rate(ws) to B.
4. The communication between B and ws is asynchronous with the time-out, t_o. After t_o, B does not wait for Rate(ws) anymore. B must pay Fee(ws) to successful ws who returns Rate(ws) within t_o.
5. B sends Min(Rate) to C.
6. If C accepts Min(Rate), C pays Fee(B) to B. Otherwise B cannot charge Fee(B) on C.
GSPN Representation

Optimization

- Optimization questions
  - \( n \): How many web services are optimal for \( B \)?
  - \( \text{Fee}(B) \): How much is an appropriate charge to customers?
  - …
- These are typically not handled well in web services discovery or composition tools
- The problem can be solved from QoS point of view in web service composition
- But this is usually segregated from “Modeling”
Selecting optimal $n$ (by HPSim)

With $n=4$, $\text{Fee}(b) = $16
$3,373$ in $10,000$ hours

Service charge, $\text{Fee}(B)$

Type Definitions

**Definition**

```
<<message Content>>
Message1

<<data>>
CFP
+Name:string
+Car_Model:string
+Loan:double

<<message Content>>
Message2

<<data>>
Propose
+Loan_Interest:double

<<message Content>>
Message3

<<data>>
Confirm
+Proposal
+Accept:Boolean

<<message Content>>
Message4

<<data>>
Inform
+Inform:string

<<interface>>
Interface CB

<<protocol>>
Protocol_CB

<<role>>
Customer

<<use>>
Interface CB

<<role>>
Broker

<<use>>
Interface CB_Callback

<<use>>
Interface CB_Callback

<<interface>>
Confirm_Proposal ([in] Message7)

<<interface>>
Propose ([in] Message6)

<<interface>>
Inform ([in] Message8)
```
**Partner Link Types**

- **Definition**
- **<<import>>**
  - **<<process>>**
    - **<<role>>**
      - **Protocol_CB::Customer**
      - **Protocol_BL1::Loan1**
      - **Protocol_BL2::Loan2**
      - **Protocol_BL3::Loan3**
      - **Protocol_BL4::Loan4**
- **<<port>>**
  - **+ Customer**
  - **+ Loan1**
  - **+ Loan2**
  - **+ Loan3**
  - **+ Loan4**

**Activity Diagram I**

- **<<receive>>**
  - **customerInput**
    - **entry/CFP(message1)**
- **<<invoke>>**
  - **Loan1Invoke**
    - **entry/Request_Interest(message1)**
- **<<receive>>**
  - **Loan1Receive**
    - **entry/Inform_Interest(message2)**
- **<<invoke>>**
  - **Loan2Invoke**
    - **entry/Request_Interest(message1)**
    - **entry/Request_Interest(message1)**
    - **entry/Request_Interest(message1)**
- **<<receive>>**
  - **Loan2Receive**
    - **entry/Inform_Interest(message3)**
    - **entry/Inform_Interest(message3)**
    - **entry/Inform_Interest(message3)**
    - **entry/Inform_Interest(message3)**
- **<<invoke>>**
  - **Loan3Invoke**
    - **entry/Request_Interest(message1)**
    - **entry/Request_Interest(message1)**
    - **entry/Request_Interest(message1)**
- **<<receive>>**
  - **Loan3Receive**
    - **entry/Inform_Interest(message4)**
    - **entry/Inform_Interest(message4)**
    - **entry/Inform_Interest(message4)**
    - **entry/Inform_Interest(message4)**
- **<<invoke>>**
  - **Loan4Invoke**
    - **entry/Request_Interest(message1)**
    - **entry/Request_Interest(message1)**
    - **entry/Request_Interest(message1)**
- **<<receive>>**
  - **Loan4Receive**
    - **entry/Inform_Interest(message5)**

- **flow**
  - **1**
Activity Diagram II

1. **InterestAssign**
   - message6/LoanInterest := message3/LoanInterest
   - [message6/LoanInterest > message3/LoanInterest]
   - switch

2. **Loan2Assign**
   - message6/LoanInterest := Message3/LoanInterest
   - [otherwise]

3. **Loan3Assign**
   - message6/LoanInterest := Message4/LoanInterest
   - [otherwise]

4. **InterestAssign**
   - message6/LoanInterest := message2/LoanInterest
   - [otherwise]

Activity Diagram III

2. **Loan4Assign**
   - message6/LoanInterest := Message5/LoanInterest
   - [message6/LoanInterest > message5/LoanInterest]
   - switch

   - **Customer**
     - entry/Propose
       - entry/Propose(message6)
     - entry/ConfirmProposal
       - entry/ConfirmProposal(message7)
     - entry/Inform
       - entry/Inform(message8)
   - [otherwise]
Conclusion

- The temporal and functional analysis for web service based software systems can increase productivity and reliability of Business Service Networks.
- Web services optimization can be benefited when tightly integrated into (semantic and graphical) modelling stage.
- Details: http://nike.psu.edu/atherton/